

PATENT

Attorney Docket No.: 25JS-156728

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of: Anne Marie HOLLER et al.	Examiner: Cristina O. Sherr
Serial No.: 10/005,729	Art Unit: 3685
Filed: November 6, 2001	
Title: OPTIMIZED SERVER FOR STREAMED APPLICATIONS	

**APPEAL BRIEF**

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This brief is in support of a Notice of Appeal filed May 25, 2010. The present paper is Appellants' Appeal Brief submitted in compliance with 37 C.F.R. 41.37(c) to the Board of Patent Appeals and Interferences to appeal the decision of Examiner Sherr in a Final Office Action dated January 1, 2010, in which pending claims 1-54 stand in final rejection. Filed herewith is a three-months extension of time, thereby extending the deadline to October 25, 2010. Accordingly, this brief is timely filed.

**REAL PARTY IN INTEREST**

Appellant in the present appeal is Endeavors Technologies, Inc.

**RELATED APPEALS AND INTERFERENCES**

Appellant and his undersigned representative are unaware of any related appeals, interferences, or judicial appearances that are concluded, ongoing, or otherwise prospective as of the date of submission of this Appeal Brief.

**STATUS OF CLAIMS**

The application was initially filed with 54 claims. Claims 1-54 are pending. Claims 2-4, 10, 18, 20-23, 28-36, 38-41, and 46-54 are withdrawn. Claims 1, 6-9, 19, 24-27, 37, and 42-45 stand rejected. The final rejection of pending claims 1-54, as presented in Appendix A, is appealed.

**STATUS OF AMENDMENTS**

There have been no amendments since the Final Office Action dated January 25, 2010 ("the Office Action").

**GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

1. Whether independent claims 1, 19, and 37 are unpatentable under 35 U.S.C. 103(a) over U.S. Patent 6,757,894 to Eylon et al. ("Eylon") in view of U.S. Patent 6,687,745 to Franco et al. Specifically, whether an Affidavit filed on September 30, 2009, under 37 CFR 1.131 is effective to swear behind Eylon.

**SUMMARY OF CLAIMED SUBJECT MATTER**

Independent claim 1 describes "a process for the optimized delivery of server based streamed applications and data to a client and the management of said streamed applications on a server" (pg. 3, paragraph [0077]). The process includes "providing application set storage means for persistently storing streamed application program sets on a said server;" (FIG. 6B, 624) "wherein said streamed application sets contain streamed application file pages;" (FIG. 4, 406) "wherein said streamed application file pages are read only;" (pg. 12, paragraph [0216]). The process includes "providing means for receiving client requests for streamed application file pages;"

(FIG. 13, 1307). The process includes "providing validation means for validating whether a client has access privilege to a requested streamed application file page;" (pg. 5, paragraph [0113]). The process includes "providing caching means for storing commonly accessed streamed application file pages in a cache;" (pg. 4, paragraph [0083]) "wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;" (pg. 6, paragraph [0137]) "wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages;" (pg. 10, paragraph [0184]). The process includes "providing compression means for compressing said requested streamed application file page before sending said requested streamed application file page to said client." (FIGS. 7A and 7B , Compression/Compressed A-D).

Independent claim 19 describes "an apparatus for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server" (pg. 3, paragraph [0077]). The apparatus includes "application set storage means for persistently storing streamed application program sets on a server;" (FIG. 6B, 624) "wherein said streamed application sets contain streamed application file pages;" (FIG. 4, 406) "wherein said streamed application file pages are read only;" (pg. 12, paragraph [0216]). The apparatus includes "means for receiving client requests for streamed application file pages;" (FIG. 13, 1307). The apparatus includes "validation means for validating whether a client has access privilege to a requested streamed application file page;" (pg. 5, paragraph [0113]). The apparatus includes "caching means for storing commonly accessed streamed application file pages in a cache;" (pg. 4, paragraph [0083]) "wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;" (pg. 6, paragraph [0137]) "wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages;" (pg. 10, paragraph [0184]). The apparatus includes "means for sending said requested streamed application file page to said client." (FIG. 6B , 621).

Independent claim 37 describes "program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform a method for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server" (pg. 3, paragraph [0077]). The method includes "providing application set storage means for persistently storing streamed application program sets on a said server;" (FIG. 6B, 624) "wherein said streamed application sets contain streamed application file pages;" (FIG. 4, 406) "wherein said streamed application file pages are read only;" (pg. 12, paragraph [0216]). The process includes "providing means for receiving client requests for streamed application file pages;" (FIG. 13, 1307). The process includes "providing validation means for validating whether a client has access privilege to a requested streamed application file page;" (pg. 5, paragraph [0113]). The process includes "providing caching means for storing commonly accessed streamed application file pages in a cache;" (pg. 4, paragraph [0083]) "wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;" (pg. 6, paragraph [0137]) "wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages;" (pg. 10, paragraph [0184]). The process includes "providing means for sending said requested streamed application file page to said client." (FIG. 6B , 621).

**ARGUMENT**

The Examiner has imposed a burden on proof of conception for an affidavit filed under 37 CFR 1.131 ("the Affidavit") that is not supported by law. The Examiner asserts on page 3 of the Office Action:

Applicant's Affidavit Table 1 recites, inter alia, an "Estream client". The Estream client is not found in the specification. Applicant's Table 1 recites ECM-LSM interaction, where ECM and LSM are not found in the specification. Specifically, the language of the claims in this case recites various means, such as "storage means", "means for receiving", "validation means", "caching means". The Affidavit refers to these means specifically as, for example, an "Estream client". The said Estream client is not found in the specification or in the claims of this case.

The Examiner has misstated the legal requirements of proving conception.

The Appellants had a prototype that was referred to as including an "Estream client," and "ECM-LSM interaction," which are associated with a physical embodiment. "The primary meaning of the word 'invention' in the Patent Act unquestionably refers to the inventor's conception rather than to a physical embodiment of that idea." *Pfaff v. Wells Electronics, Inc.*, 525 U.S. 55, 48 USPQ2d 1641 (1998). As the Pfaff Court indicates, the 'invention' of the claims refers to the conception, not the physical embodiment thereof. Therefore, the Affidavit should show evidence of conception, which may or may not make reference to physical embodiments. The Pfaff Court would find repugnant a requirement regarding the names of components that could change for reasons that have nothing to do with the functionality of the invention (e.g., for marketing reasons). The Affidavit includes Table 1, which correlates each and every element of the claims with a description of a physical embodiment of the invention as of a date prior to the cited reference. The claim elements use functional language that reads on the function of the physical embodiments illustrated by the evidence. The conception of that functionality is distinct from the physical embodiments that illustrate it by way of example. The Examiner's requirement that the Affidavit do more than establish conception, by using language that is also used in the specification and the claims, is without legal merit.

While the Affidavit is relatively complete in showing prior conception of each and every element of claim 1, such a complete showing is not required by law. In *Hybritech Inc. v. Monoclonal*

*Antibodies, Inc.*, 802 F.2d 1367, 231 USPQ 81 (Fed. Cir. 1986), cert. denied, 480 U.S. 947 (1987), the Court found conception was adequately shown even though lab notebooks did not expressly state the liters/mole limitation [of the claims]. If it is the case that the physical embodiment illustrated by way of example in the Affidavit addresses each and every element of claim 1, then the Affidavit includes evidence that is more complete than the lab notebooks considered by the Hybritech Court, and which the Hybritech Court concluded were adequate to show conception. (Note: Although evidentiary support is not required for every feature, conception must include every feature. See, e.g., *Kridl v. McCormick*, 105 F.3d 1446, 41 USPQ2d 1686 (Fed. Cir. 1997).)

The evidence provided in the Affidavit is not required to be enabling. Regardless of whether the Affidavit appears to illustrate an enabling embodiment of the invention, the issue is conception, not enablement. In *Fiers v. Revel*, 984 F.2d 1164, 25 USPQ2d 1601 (Fed. Cir. 1993), the Court stated: "Since Fiers seeks to establish priority under section 102(g), the controlling issue here is whether he conceived a DNA coding for beta-IF, not whether his method was enabling." The Examiner's requirement that the "Estream client" and other components be referenced word-for-word in the specification and claims has no legal merit. The evidence provided in the Affidavit is only required to show conception. (Appellants are not suggesting that the evidence is not enabling, only that it is not **required** to be enabling. Appellants are not required to opine on whether the evidence is enabling because taking a position would have no relevance in a determination as to whether the Affidavit proves conception.)

The Examiner states on page 3 of the Office Action that since the Estream client term does not appear in the specification or in the claims, the content or facts of the Affidavit do not comport and are not commensurate with the disclosure of the instant application. ***The Examiner's assertion is demonstrably false.*** The claims include language that reads directly on components of the physical embodiment. The language specifically identified by the Examiner as having no supporting evidence is addressed individually below for "storage means," "means for receiving," "validation means," and "caching means."

Storage Means

Regarding the specific language the Examiner identified in the Office Action, in Table 1, page 4 of the Affidavit, "providing application set storage means for persistently storing streamed application program sets on a server" is recited with evidentiary support. The evidence includes in part:

C7) Estream client-server diagram comprising an application server. Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client.

C9) Application stream set builder (pg. 12)

The application stream is build on a computer.

The particular information, or at least an obvious variant, associated with persistently storing streamed application program sets on a server is disclosed. Specifically, if the application stream is built on a computer, the application stream can be characterized as the means for persistently storing streamed application programs (or program sets). Although the Examiner asserts that the Estream server is not explicitly mentioned in the specification, the Examiner provides no support for such a requirement and Appellants believe no such requirement exists in the law. Moreover, the application server, regardless of whether it is referred to as an "Estream server," is used for streaming application programs across a computer network while executing application programs on a client, which would require persistent storage on the server. So, in addition to the Examiner's assertion having no basis in law, the evidence of conception clearly illustrates "providing application set storage means for persistently storing streamed application program sets on a server" or an obvious variant thereof.

Claim 19 includes the language "application set storage means for persistently storing streamed application program sets on a server." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Claim 37 includes the language "providing application set storage means for persistently storing streamed application program sets on said server." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Means for Receiving

Regarding the specific language the Examiner identified in the Office Action, in Table 1, page 5 of the Affidavit, "providing means for receiving client request for streamed application file pages" is recited with evidentiary support. The evidence includes in part:

C9) An application file server responds to requests by client application cache manager for portions of application's files (pg. 12).

C16) Each server responds to a single self-contained message from a client requesting a page set from the server (pg. 1).

C18) Application Server (pg. 26)

The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.

The particular information, or at least an obvious variant, associated with receiving requests from a client at a server is disclosed. Specifically, if the application server receives requests from a client, then the application server must necessarily include a means for receiving the request. Although the Examiner asserts that the Estream server is not explicitly mentioned in the specification, the Examiner provides no support for such a requirement and Appellants believe no such requirement exists in the law. Moreover, the application server, regardless of whether it is referred to as an "Estream server," receives requests from a client, which would require a means for receiving the requests. So, in addition to the Examiner's assertion having no basis in law, the evidence of conception clearly illustrates "providing means for receiving client request for streamed application file pages" or an obvious variant thereof.

Claim 19 includes the language "means for receiving client requests for streamed application file pages." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Claim 37 includes the language "providing means for receiving client requests for streamed application file pages." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.



Validation Means

Regarding the specific language the Examiner identified in the Office Action, in Table 1, page 6 of the Affidavit, "providing validation means for validating whether a client has access privilege to a requested streamed application file page" is recited with evidentiary support. The evidence includes in part:

## C3) ECM-LSM Interaction (pg. 9)

When ECM receives a request, it checks to determine if an access token is available.

## C5) Execution of application (pg. 2)

Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.

Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data.

## DRM server (pg. 2)

DRM server determines authentication of a client, validates application licenses and so on.

## C8) Accessing Files (pg. 3)

The client would provide the file id of interest and the proper access token for a file.

The system verifies it.

## C16) Security (pg. 9)

There are two levels of security involved in the AS. One of them is to require the client to have an AccessToken and presents it to the AS upon every request.

The particular information, or at least an obvious variant, associated with validating the access privilege of a client for a requested streamed application file page is disclosed. Specifically, if the application server validates a client request for access to the streamed application, the server necessarily includes a validation means for validating whether a client has access privilege for the requested streamed application. Although the Examiner asserts that the ECM-LSM is not

explicitly mentioned in the specification, the Examiner provides no support for such a requirement and Appellants believe no such requirement exists in the law. Moreover, the application server, regardless of whether it is referred to as part of an "ECM-LSM" system validates requests from a client, which would require a means for validating the requests. So, in addition to the Examiner's assertion having no basis in law, the evidence of conception clearly illustrates "providing validation means for validating whether a client has access privilege to a requested streamed application file page" or an obvious variant thereof.

Claim 19 includes the language "validation means for validating whether a client has access privilege to a requested streamed application file page." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Claim 37 includes the language "providing validation means for validating whether a client has access privilege to a requested streamed application file page." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

#### Caching Means

Regarding the specific language the Examiner identified in the Office Action, in Table 1, page 7 of the Affidavit, "providing caching means for storing commonly accessed streamed application file pages in a cache" is recited with evidentiary support. The evidence includes in part:

##### C3) Functionality (pg. 1)

The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3).

Diagram illustrating overall client architecture comprising cache elements. (pg. 8).

Implementation of the cache manager. (pgs. 11-17).

##### C4) Cache organization (pg. 1)

The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.

C9) The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application

and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manager for the desired data. (pg. 13, 4<sup>th</sup> full paragraph).

The particular information, or at least an obvious variant, associated with caching streamed application file pages in a cache is disclosed. Specifically, if the application pages are cached at a client, then the client necessarily includes a caching means for caching accessed streamed application file pages in a cache. So, in addition to the Examiner's assertion having no basis in law, the evidence of conception clearly illustrates "providing caching means for storing commonly accessed streamed application file pages in a cache" or an obvious variant thereof.

Claim 19 includes the language "caching means for storing commonly accessed streamed application file pages in a cache." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Claim 37 includes the language "providing caching means for storing commonly accessed streamed application file pages in a cache." This language has adequate evidentiary support, which is provided in Table 1 of the Affidavit in association with claim 1.

Appellants show possession of every feature recited in the claims (and therefore, necessarily, every feature of Eylon used to formulate a rejection of the claims) in a physical embodiment and prove by corroborating evidence, included in the Affidavit, that the inventor disclosed a completed thought in such clear terms as to enable those skilled in the art to make the invention. Therefore, Appellants have met their burden of showing evidence of prior conception with the Affidavit. The additional requirements imposed by the Examiner are not required by law.

**CLAIMS APPENDIX**

Provided in Appendix A, attached.

**EVIDENCE APPENDIX**

None.

**RELATED PROCEEDINGS APPENDIX**

None.

**Conclusion**

Appellants show possession of every feature recited in the claims (and therefore, necessarily, every feature of Eylon used to formulate a rejection of the claims) in a physical embodiment and prove by corroborating evidence, included in the Affidavit, that the inventor disclosed a completed thought in such clear terms as to enable those skilled in the art to make the invention. Therefore, Appellants have met their burden of showing evidence of prior conception with the Affidavit. The additional requirements imposed by the Examiner are not required by law.

Respectfully submitted,

Date: October 25, 2010

By: William F. Ahmann  
William F. Ahmann  
Reg. No. 52,548  
Customer Number 69849  
Sheppard Mullin Richter & Hampton LLP  
990 Marsh Road  
Menlo Park, CA 94025  
Telephone: (650) 815-2613

**CERTIFICATE OF ELECTRONIC TRANSMISSION**

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being filed electronically via EFS-Web addressed to the Commissioner for Patents on:

Date: 10/25/10 By: Valerie Cohen  
Valerie Cohen

## APPENDIX A (CLAIMS APPENDIX)

1. A process for the optimized delivery of server based streamed applications and data to a client and the management of said streamed applications on a server, the process comprising:

providing application set storage means for persistently storing streamed application program sets on a said server;

wherein said streamed application sets contain streamed application file pages;

wherein said streamed application file pages are read only;

providing means for receiving client requests for streamed application file pages;

providing validation means for validating whether a client has access privilege to a requested streamed application file page;

providing caching means for storing commonly accessed streamed application file pages in a cache;

wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;

wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages; and

providing compression means for compressing said requested streamed application file page before sending said requested streamed application file page to said client.

2. The process of claim 1, further comprising:

providing compression means for compressing said requested streamed application file page before sending said requested streamed application file page to said client.

3. The process of claim 2, wherein when a client requests multiple streamed application file pages, said server concatenates all of the requested pages and compresses the entire set at once before sending to said client.
4. The process of claim 2, wherein said commonly accessed streamed application file pages are compressed before being stored in said cache.
5. The process of claim 1, wherein all of the streamed application file pages in the streamed application sets are compressed before being stored in said application set storage means.
6. The process of claim 1, further comprising:  
providing profiling means for profiling the access patterns of streamed application file pages.
7. The process of claim 6, wherein said access patterns are sent to said client to guide its prefetching of streamed application file pages.
8. The process of claim 6, wherein said access patterns are used by said server to pre-package and compress groups of streamed application file pages;  
and wherein a prepackaged group is sent to a client requesting pages within a set.
9. The process of claim 6, wherein said access patterns are used by said server to perform prefetching of streamed application file pages for pushing to clients.
10. The process of claim 1, further comprising:  
providing a license server;  
wherein said validation means resides on said license server; and  
wherein said validation means provides a client with an access token that contains information regarding access rights, the application that it applies to, and an expiration time.
11. The process of claim 10, further comprising:

providing token reception means on said server for receiving an access token from a client;

providing decryption means on said server for decrypting said access token;

providing means for validating the contents of said access token; and granting access to a client with a valid access token.

12. The process of claim 10, wherein said decrypting means uses a secret key shared with said license server to decrypt an access token.

13. The process of claim 10, wherein said server maintains a list of recently approved access tokens and compares incoming access tokens with said list, and wherein incoming access tokens that match an entry on said list are approved without further processing.

14. The process of claim 1, wherein a client uses a persistent connection over the Internet with said server to make multiple requests from said server, and wherein said server closes persistent connections that have been idle for a predetermined period of time.

15. The process of claim 1, further comprising:

assigning individual servers a specific set of streamed applications sets to serve to clients;

and

wherein the servers across a network are asymmetrically assigned different sets of streamed application sets to improve overall server efficiency.

16. The process of claim 15, wherein said individual servers are dynamically assigned streamed application sets to match client accesses over time.

17. The process of claim 15, further comprising:

providing a central control server;



wherein said individual servers periodically send a summary of their file access patterns to said central control server; and

wherein said central control server reassigns individual servers according to the file access patterns.

18. The process of claim 1, wherein said server communicates with clients across the Internet.

19. An apparatus for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server, the apparatus comprising:

application set storage means for persistently storing streamed application program sets on a server;

wherein said streamed application sets contain streamed application file pages;

wherein said streamed application file pages are read only;

means for receiving client requests for streamed application file pages;

validation means for validating whether a client has access privilege to a requested streamed application file page;

caching means for storing commonly accessed streamed application file pages in a cache;

wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;

wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages; and

means for sending said requested streamed application file page to said client.

20. The apparatus of claim 19, further comprising:

compression means for compressing said requested streamed application file page before sending said requested streamed application file page to said client.

21. The apparatus of claim 20, wherein when a client requests multiple streamed application file pages, said server concatenates all of the requested pages and compresses the entire set at once before sending to said client.

22. The apparatus of claim 20, wherein said commonly accessed streamed application file pages are compressed before being stored in said cache.

23. The apparatus of claim 19, wherein all of the streamed application file pages in the streamed application sets are compressed before being stored in said application set storage means.

24. The apparatus of claim 19, further comprising:

profiling means for profiling the access patterns of streamed application file pages.

25. The apparatus of claim 24, wherein said access patterns are sent to said client to guide its prefetching of streamed application file pages.

26. The apparatus of claim 24, wherein said access patterns are used by said server to pre-package and compress groups of streamed application file pages; and wherein a pre-packaged group is sent to a client requesting pages within a set.

27. The apparatus of claim 24, wherein said access patterns are used by said server to perform prefetching of streamed application file pages for pushing to clients.

28. The apparatus of claim 19, further comprising:

a license server;

wherein said validation means resides on said license server; and

wherein said validation means provides a client with an access token that contains information regarding access rights, the application that it applies to, and an expiration time.

29. The apparatus of claim 28, further comprising:

token reception means on said server for receiving an access token from a client;

decryption means on said server for decrypting said access token; means for validating the contents of said access token; and

a module for granting access to a client with a valid access token.

30. The apparatus of claim 28, wherein said decrypting means uses a secret key shared with said license server to decrypt an access token.

31. The apparatus of claim 28, wherein said server maintains a list of recently approved access tokens and compares incoming access tokens with said list, and wherein incoming access tokens that match an entry on said list are approved without further processing.

32. The apparatus of claim 19, wherein a client uses a persistent connection over the Internet with said server to make multiple requests from said server, and wherein said server closes persistent connections that have been idle for a predetermined period of time.

33. The apparatus of claim 19, further comprising:

a module for assigning individual servers a specific set of streamed applications sets to serve to clients; and

wherein the servers across a network are asymmetrically assigned different sets of streamed application sets to improve overall server efficiency.

34. The apparatus of claim 33, wherein said individual servers are dynamically assigned streamed application sets to match client accesses over time.

35. The apparatus of claim 33, further comprising:

providing a central control server;

wherein said individual servers periodically send a summary of their file access patterns to said central control server; and

wherein said central control server reassigns individual servers according to the file access patterns.

36. The apparatus of claim 19, wherein said server communicates with clients across the Internet.

37. A program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform a method for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server, the method comprising:

providing application set storage means for persistently storing streamed application program sets on a said server;

wherein said streamed application sets contain streamed application file pages;

wherein said streamed application file pages are read only;

providing means for receiving client requests for streamed application file pages;

providing validation means for validating whether a client has access privilege to a requested streamed application file page;

providing caching means for storing commonly accessed streamed application file pages in a cache;

wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;

wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages; and

providing means for sending said requested streamed application file page to said client.

38. The method of claim 37, further comprising:

providing compression means for compressing said requested streamed application file page before sending said requested streamed application file page to said client.

39. The method of claim 38, wherein when a client requests multiple streamed application file pages, said server concatenates all of the requested pages and compresses the entire set at once before sending to said client.

40. The method of claim 38, wherein said commonly accessed streamed application file pages are compressed before being stored in said cache.

41. The method of claim 37, wherein all of the streamed application file pages in the streamed application sets are compressed before being stored in said application set storage means.

42. The method of claim 37, further comprising:

providing profiling means for profiling the access patterns of streamed application file pages.

43. The method of claim 42, wherein said access patterns are sent to said client to guide its prefetching of streamed application file pages.

44. The method of claim 42, wherein said access patterns are used by said server to pre-package and compress groups of streamed application file pages; and wherein a pre-packaged group is sent to a client requesting pages within a set.
45. The method of claim 42, wherein said access patterns are used by said server to perform prefetching of streamed application file pages for pushing to clients.
46. The method of claim 37, further comprising:  
providing a license server;  
wherein said validation means resides on said license server; and  
wherein said validation means provides a client with an access token that contains information regarding access rights, the application that it applies to, and an expiration time.
47. The method of claim 46, further comprising:  
providing token reception means on said server for receiving an access token from a client;  
providing decryption means on said server for decrypting said access token;  
providing means for validating the contents of said access token; and granting access to a client with a valid access token.
48. The method of claim 46, wherein said decrypting means uses a secret key shared with said license server to decrypt an access token.
49. The method of claim 46, wherein said server maintains a list of recently approved access tokens and compares incoming access tokens with said list, and wherein incoming access tokens that match an entry on said list are approved without further processing.

50. The method of claim 37, wherein a client uses a persistent connection over the Internet with said server to make multiple requests from said server, and wherein said server closes persistent connections that have been idle for a predetermined period of time.

51. The method of claim 37, further comprising:

assigning individual servers a specific set of streamed applications sets to serve to clients;

and

wherein the servers across a network are asymmetrically assigned different sets of streamed application sets to improve overall server efficiency.

52. The method of claim 51, wherein said individual servers are dynamically assigned streamed application sets to match client accesses over time.

53. The method of claim 51, further comprising:

providing a central control server;

wherein said individual servers periodically send a summary of their file access patterns to said central control server; and

wherein said central control server reassigns individual servers according to the file access patterns.

54. The method of claim 37, wherein said server communicates with clients across the Internet.